II International Conference on Particle-based Methods - Fundamentals and Applications
PARTICLES 2011
E. Oñate and D.R.J. Owen (Eds)

DYNAMIC REFINEMENT FOR FLUID FLOW SIMULATIONS WITH SPH

YAIDEL REYES LÓPEZ* AND DIRK ROOSE†

*Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A - bus 2402. 3001 Heverlee, Leuven, Belgium.

Centro de Investigaciones de Métodos Computacionales y Numéricos
en la Ingeniería (CIMCNI),
Universidad Central "Marta Abreu" de Las Villas (UCLV)
Road to Camajuaní Km. $5\frac{1}{2}$, 54830 Santa Clara, Villa Clara, Cuba.
e-mail: yaidel.reyeslopez@cs.kuleuven.be

†Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A - bus 2402. 3001 Heverlee, Leuven, Belgium.
e-mail: dirk.roose@cs.kuleuven.be

Key words: particle methods, SPH, dynamic refinement, adaptivity

Abstract. In this paper, we present a dynamic refinement algorithm for the SPH method where a particle is refined by replacing it with smaller daughter particles. The position of the new particles is calculated by using a square pattern centered at the position of the refined particle. We propose to reduce the error introduced by the refinement by determining the separation of the pattern and the smoothing distance of the daughter particles such that the kernel gradient error is minimized. The results of the simulations using the fully refined domain and the simulations using the dynamic refinement starting from the unrefined domain are compared and are in a good agreement. Better results are obtained when the proposed method to reduce the error is used.

1 INTRODUCTION

Smoothed Particle Hydrodynamics (SPH) is a fully mesh-free Lagrangian method that has been successfully applied in several types of problems including fluid dynamics and deformation of solids. Using SPH the domain is discretized by a set of particles (or interpolation points) that independently carry the material properties and the local resolution or smoothing distance (h) of the particle. Varying the smoothing length allows to vary the spatial resolution. Nevertheless, it is not common in SPH simulations to have different levels of refinement, as in e.g. the Finite Element Method (FEM), to improve the accuracy and/or to reduce computational cost.

The first steps in dynamic adaptivity in SPH were performed for astrophysical simulations, where the density was used as the criterion to change the resolution [1, 5, 6, 11, 14]. In other fields Lastiwka and co-workers [7] proposed a method for adaptively inserting and removing particles and they tested the method in a shock tube problem. More recently, Feldman and Bonet [4] proposed a particle refinement procedure, where daughter particles are located using axis aligned hexagonal and triangular patterns that are scaled according to a spread parameter defined before the simulation starts.

In this paper we present a dynamic refinement procedure in which particles are split based on a velocity criterion. The new "daughter" particles are located in a square pattern centered at the position of the refined particle. The optimal separation of the pattern and the smoothing distance of the daughter particles are determined by minimizing the error produced by the refinement in the gradient of the kernel.

2 SPH Background

The SPH method provides a numerical solution for integral equations and partial differential equations (PDEs). Comprehensive reviews of the method can be found in [8, 9, 12]. Herein we only present the formulation used in this work.

2.1 SPH for general fluid dynamics

Using SPH, the particle approximation of a function is given by

$$\langle f(\mathbf{x})\rangle = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) W(\mathbf{x}-\mathbf{x}_j, h), \tag{1}$$

where m_j and ρ_j are the mass and the density of the particle j respectively, and $\frac{m_j}{\rho_j}$ approximates the volume of the particle. To approximate the gradient of a function, the following expression is used

$$\langle \nabla f(\mathbf{x})\rangle = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) \nabla W(\mathbf{x}-\mathbf{x}_j, h). \tag{2}$$

The smoothing length h defines the support domain of the kernel W. Major properties and requirements of W are summarized and described in [8, 9]. For the simulations presented in this work, a piecewise cubic spline is used as the kernel function.

The SPH approximation, to be applied in general fluid dynamics, is derived by discretizing the Navier-Stokes equations in Lagrangian form. The particle approximation of the density (ρ) is obtained according to the continuity equation using the particle approximation of the velocity gradient plus some transformations, obtaining

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^{N} m_j (v_i^\alpha - v_j^\alpha) \frac{\partial W_{ij}}{\partial x_i^\alpha}, \tag{3}$$

where v and x are the velocity and position respectively (the Einstein notation is applied).

Similarly, the particle approximation of momentum evolution is derived by approximating the stress gradient in the Navier-Stokes momentum equation and applying some transformations. If the viscous part of the stress tensor is neglected and only the pressure is retained then it results in

$$\frac{Dv_i^\alpha}{Dt} = -\sum_{j=1}^{N} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial x^\alpha} + F_i^\alpha, \tag{4}$$

where p denotes pressure and F_i^α represents the external forces.

Although we keep the value of h fixed for each particle, the dynamic refinement procedure introduces particles with smaller smoothing distance to increase the local resolution. When particles with different smoothing distances interact, it is necessary to symmetrize the interaction between them to avoid a violation of Newton's third low. Different approaches for preserving the symmetry have been used (see [8]). In this work the smoothing distance used to compute the interaction between particles is modified to produce a symmetric smoothing length using the arithmetic mean. This is, $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h_{ij})$, with $h_{ij} = (h_i + h_j)/2$. In the next sections the notation $W_j(\mathbf{x}) = W(\mathbf{x} - \mathbf{x}_j, \frac{h(\mathbf{x})+h_j}{2})$ is adopted.

3 REFINEMENT PROCEDURE

In SPH simulations, the finest resolution needed to obtain the required accuracy is commonly used in the whole domain. However, often zones are observed where the flow of the particles is slower and the properties behave smoother. In these zones, fewer particles can be used while a similar accuracy can be achieved. To exploit this, a coarse domain discretization is used at the beginning of the simulation. During the simulation the zones where more particles are needed are identified by using a refinement criterion, and the resolution is locally increased in these zones by applying a refinement procedure.

Refinement is useful only if: (a) there is a gain in execution time compared with the simulation using the fully refined domain; (b) local properties are only slightly changed and global properties like kinetic energy, and linear and angular momentum are conserved; (c) the results obtained using the refinement are an improvement over the results of the simulation using the unrefined domain. Notice that for dynamic refinement the additional computational cost introduced by the refinement procedure has to be carefully considered, since both the identification of the particles to be refined and the procedure to increase the resolution at these particles are performed during the simulation loop.

3.1 Refinement criterion

Several criteria can be used to determine which particles should be refined during the simulation, depending on the type of the problem. Feldman and Bonet [4] used refinement zones, splitting all the particles that move inside those zones. The number of neighbors can also be used and particles with few neighbors can be split to maintain

the local accuracy. Physical properties have been successfully used too as criteria for the refinement. Kitsionas and Whitworth [5, 6] applied a refinement criterion based on a physical requirement known as "Jeans Condition" and Lastiwka et al. [7] used a criterion based on the velocity gradient.

For the simulations in this work, a criterion based on the velocity of the particles is used for dynamic refinement. Particles with a velocity greater than a pre-established threshold are refined. This refinement criterion was found to be adequate in the problems that we considered. Note that the refinement algorithm described below is totally independent of the refinement criterion.

3.2 Refinement procedure

The refinement procedure presented in this work is based on particle splitting, i.e. if one particle n is selected to be refined, it is replaced by the daughter particles $d = 1, ..., M$. The properties of these newly introduced particles have to be assigned in a convenient way. Using the continuity density approach, the properties to be assigned are $(\mathbf{x}_d, \mathbf{v}_d, m_d, \rho_d, h_d, \mathbb{O})$, where $\mathbb{O}$ represents a set of other properties that depends on the model. Unless there is a physical law that imposes a restriction or condition, the properties should be interpolated to guarantee a smooth representation of the current status of the system in the neighborhood where the refinement takes place.

In this work the position of the daughter particles is calculated by using a square pattern centered at the position of the refined particle. In [15] we studied the possibility of rotating and scaling the square according to the local distribution of the particles in order to reduce the overlap of the newly created daughter particles with the neighbors. This approach introduced additional "damping" in some simulations, and will not be used in this paper. In the next sections we derive an error estimate and a proposal to minimize it.

Similarly to Feldman and Bonet [3, 4], in this work we define two refinement parameters: the separation parameter $\epsilon \in (0, 1)$ and the smoothing ratio $\alpha \in (0, 1)$. The first determines the spread of the pattern, i.e. daughter particles are placed in a square with side ϵd_{ref}, where d_{ref} is a reference distance, taken to be the initial inter-particle space. The smoothing ratio α relates the smoothing distances of the daughter particles h_d and of the original particle h_n by $h_d = \alpha h_n$. To determine the final position, the rotation of the pattern has to be decided. In the results presented in this work we use an axis aligned orientation, but a randomly rotated pattern can be used as well.

It was showed in [3, 4] that the only velocity distribution that conserves both linear momentum and kinetic energy is obtained by copying the velocity of the refined particle to its daughters. In addition, a symmetric distribution of the daughter particles and their masses about the original particle position also conserves the angular momentum. In order to preserve these global properties, the velocity of the refined particle is copied to its daughters and the mass is uniformly distributed, i.e. $v_d = v_n$ and $m_d = 0.25 m_n$.

The other properties are interpolated at the daughter particle positions. For this

we adopt the Corrective Smoothed Particle Method (CSPM) [2] that provides a particle approximation that has 0th order consistency for both interior and boundary particles [9]. The interpolated properties are calculated according to

$$f(\mathbf{x}) = \frac{\sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j, h_j)}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} W(\mathbf{x} - \mathbf{x}_j, h_j)}, \tag{5}$$

where W is taken to be the cubic spline.

3.3 Refinement error

The refinement procedure modifies the local properties leading to an error in the current status of the system. It is important to have control over the error introduced in order to keep the accuracy of the simulation. In this section we derive a measure for the error using a procedure similar to that of Feldman and Bonet [3, 4], but we focus on the error introduced in the particle approximation of the gradient of a function because the gradient is used in the derivation of the SPH approximation of the continuity and the momentum equation.

Let's consider the expression given in (2), now taking into account that the symmetrized smoothing distance is used to evaluate the kernel. Although in this work the square pattern is used, we start with the derivation of the error in the general case when a particle n is refined into M daughter particles. If the particle n is refined the approximation of the gradient of a function changes to

$$\langle \nabla f(\mathbf{x}) \rangle^* = \langle \nabla f(\mathbf{x}) \rangle - \frac{m_n}{\rho_n} f(\mathbf{x}_n) \nabla W_n(\mathbf{x}) + \sum_{d=1}^{M} \frac{m_d}{\rho_d} f(\mathbf{x}_d) \nabla W_d(\mathbf{x}). \tag{6}$$

The local error produced at $\mathbf{x}$ due to the refinement of the particle n can be defined as the sum of the square of the error introduced in each direction of the gradient, i.e.

$$\begin{aligned} e_n(\mathbf{x}) &= \left(\langle \frac{\partial f(\mathbf{x})}{\partial x^\alpha} \rangle - \langle \frac{\partial f(\mathbf{x})}{\partial x^\alpha} \rangle^* \right)^2 \\ &= m_n^2 \left(\frac{f(\mathbf{x}_n)}{\rho_n} \frac{\partial W_n(\mathbf{x})}{\partial x^\alpha} - \sum_{d=1}^{M} \lambda_d \frac{f(\mathbf{x}_d)}{\rho_d} \frac{\partial W_d(\mathbf{x})}{\partial x^\alpha} \right)^2, \end{aligned} \tag{7}$$

where $m_d = \lambda_d m_n$ and $\sum_{d=1}^{M} \lambda_d = 1$ to ensure mass conservation, and α is a summation index that iterates over the component of the gradient according to the Einstein notation. Now the global error can be written as

$$E_n = \int_\Omega e_n(\mathbf{x}) d\mathbf{x}. \tag{8}$$

We will analyze this error for more specific cases.

3.4 Density refinement error using the continuity density approach

Using the continuity density approach, more specifically (3), and considering that the velocity of the daughter particles is copied from the refined particle, the local error introduced by the refinement in the approximation of the material derivative of the density is defined by

$$\begin{aligned} e_n^\rho &= \langle \frac{D\rho(\mathbf{x})}{Dt} \rangle - \langle \frac{D\rho(\mathbf{x})}{Dt} \rangle^* \\ &= m_n \left(\mathbf{v}(\mathbf{x}) - \mathbf{v}_n \right) \cdot \left(\nabla W_n(\mathbf{x}) - \sum_{d=1}^{M} \lambda_d \nabla W_d(\mathbf{x}) \right), \end{aligned} \tag{9}$$

from where the global error can be obtained as

$$E_n^\rho = m_n^2 \int_\Omega \left[\left(\mathbf{v}(\mathbf{x}) - \mathbf{v}_n \right) \cdot \left(\nabla W_n(\mathbf{x}) - \sum_{d=1}^{M} \lambda_d \nabla W_d(\mathbf{x}) \right) \right]^2 d\mathbf{x}. \tag{10}$$

Now we can define the following problem:

Problem 1. *Find* $(\mathbf{x}_d; h_d; \lambda_d)$*, with* $d = 1..M$*, such that* E_n^ρ *is minimal.*

Since E_n^ρ depends on $\mathbf{v}$, which varies in space and time, problem 1 has to be solved each time a particle is refined, which is computationally expensive. Taking into account that the value of the velocities does not change at time t when the refinement is performed, an approximation of the solution to problem 1 can be obtained by minimizing

$$E_n^{\nabla W} = \int_\Omega \left(\frac{\partial W_j(\mathbf{x})}{\partial x^\alpha} - \sum_{d=1}^{M} \lambda_d \frac{\partial W_d(\mathbf{x})}{\partial x^\alpha} \right)^2 d\mathbf{x}, \tag{11}$$

which we refer herein as the kernel gradient error.

Still $E_n^{\nabla W}$ depends on $h(x)$ that also varies in space and time due to the refinement. However, if we fix the refinement parameter α, there are only two possible values for the smoothing distance: h_0, which is the initial value set to the unrefined particles, and αh_0, assigned to the daughter particles. Considering this, the analysis of (11) can be simplified. We take into account two cases: first, $h(x) = h_0, \forall x \in I_D$, where I_D is the influence domain of the particle to be refined; and second, $h(x) = \alpha h_0, \forall x \in I_D$. The solution that gives the smallest values of $E_n^{\nabla W}$ for these two cases can be selected as a good approximation of the solution to problem 1.

3.5 Kernel gradient error using the square pattern.

Using the square pattern, $\lambda_d = 0.25$ for $d = 1..4$, we determine values for ϵ and α that give a good approximation to the solution of problem 1. Fig. 1 shows the kernel gradient error for different values of (ϵ, α) for the two considered cases. We should select values for (ϵ, α) that lead to a small error in both cases, taking into account two more elements: a) if the smoothing distance of the daughter particles is too large too many neighbors would be inside the support domain, which in turn would smooth out the properties of the daughter particles and would increase the computational cost; b) the separation parameter should be chosen to avoid clumped or spaced out distributions. For these reasons we consider that (ϵ, α) have to be chosen around (0.5,0.5). In this work, we choose $(\epsilon, \alpha) = (0.55, 0.7)$, highlighted with red points in fig. 1.

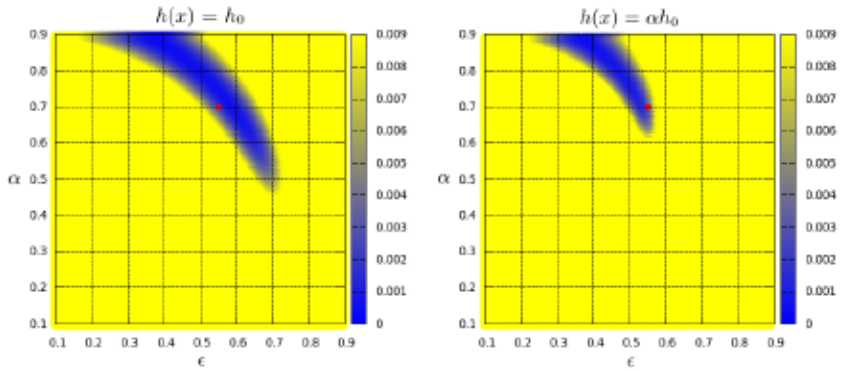

Figure 1: Kernel gradient error less than 0.009 for the two considered cases: left, $h(x) = h_0$; and right, $h(x) = \alpha h_0$. The red points show the position $(0.55, 0.7)$.

4 FREE SURFACE FLOW SIMULATIONS USING DYNAMIC REFINEMENT

To simulate free surface flows Monaghan [13] proposed to use the continuity density approach (see eq. 3) together with

$$\frac{Dv_i^\alpha}{Dt} = -\sum_{j=1}^{N} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} - \Pi_{ij} \right) \frac{\partial W_{ij}}{\partial x^\alpha} + F_i^\alpha, \tag{12}$$

to evolve the momentum. The artificial viscosity term Π_{ij} is introduced to reduce numerical instabilities that result from the absence of viscous forces in eq. 4. The artificial viscosity is calculated using

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}} & \text{if } \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} < 0 \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where $\mu_{ij} = \frac{h\mathbf{v}_{ij}\cdot\mathbf{x}_{ij}}{\mathbf{x}_{ij}^2+0.01h^2}$, $\bar{\rho}_{ij} = \frac{\rho_i+\rho_j}{2}$, the notation $A_{ij} = A_i - A_j$ is used, and α and β are constants that change with the problem.

In [13], an incompressible flow is considered to be slightly compressible. The relative fluctuation in density is proportional to $(\frac{v_{typ}}{c})^2 = M^2$, where v_{typ} is the typical bulk velocity, c is the speed of sound, and M is the Mach number. The value for c is chosen to give a density fluctuation of $\sim 1\%$ resulting in $M = 0.1$. To close the system of governing equations the pressure is calculated by applying the equation of state $p_i = B\left(\left(\frac{\rho_i}{\rho_0}\right)^\gamma - 1\right)$, where γ is a constant commonly taken as 7, ρ_0 is the reference density, and B is a problem dependent constant related with the speed of sound by $c = \sqrt{\frac{\partial p}{\partial \rho}} \approx \sqrt{\frac{7B}{\rho_0}}$.

4.0.1 Breaking dam problem

In the breaking dam problem a 25m side square block of water is considered. Initially the water is at rest in a container until the right wall is removed suddenly. No-slip boundary conditions are applied in solid boundaries using the ghost particle approach (see [8]).

For this problem we study the effect of applying dynamic refinement starting from two different initial discretizations that we call *finer* and *coarser*. Then, the results obtained applying the dynamic refinement using $(\epsilon, \alpha) = (0.5, 0.5)$ and $(\epsilon, \alpha) = (0.55, 0.7)$ are compared with the results using the corresponding unrefined and fully refined domains.

The simulations for the two different discretizations are set up as follows:

- **Finer discretization**. The unrefined domain has an initial inter-particle spacing equal to 0.833334 that correspond to 900 particles located in a lattice; the fully refined domain is obtained by statically refining the 900 particles to a finer lattice of 3600 particles with an initial inter-particle spacing of 0.416667; the simulations using dynamic refinement start from the unrefined domain and use as the refinement criterion a velocity threshold of $v_{max} = 10m/s$.

- **Coarser discretization**. In this case the unrefined domain has an initial inter-particle spacing equal to 1.666667 corresponding to a lattice of 225 particles. As in the previous case, the fully refined domain is obtained by statically refining the 225 particles to a finer lattice of 900 particles[1] with an initial inter-particle spacing of 0.833334 . The simulations using dynamic refinement start from the unrefined domain and use for the refinement criterion a velocity threshold of $v_{max} = 7m/s$ ($v_{max} = 10m/s$ gives poor results).

Fig. 2 shows the evolution of the particles during the simulations for the finer discretization using dynamic refinement with $(\epsilon, \alpha) = (0.55, 0.7)$ and using the fully refined

[1]Notice that the fully refined domain corresponding to the coarser discretization is the same as the unrefined domain of the finer discretization.

domain. We observe a good agreement in the distribution and the velocity of the particles. The same occurs in the simulation using $(\epsilon, \alpha) = (0.5, 0.5)$. If the simulations corresponding to the coarser discretization are compared among them in a similar way, the particle distribution and the velocity of the particles are in a good agreement as well.

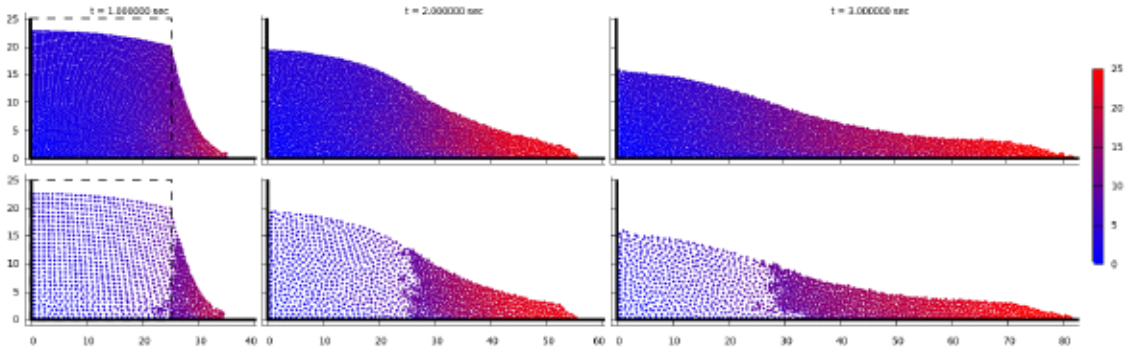

Figure 2: Breaking dam simulation.

The evolution of the surge front (Z) of the water particles is shown in Fig. 3, where also the experimental result of Martin and Moyce [10] are presented. The values of Z are normalized by the initial value $Z_0 = 25m$ and the time is normalized by $\sqrt{H_0/g}$ with g the gravity force. Contrary to what we expected, the simulation with the lowest resolution gives the results that are closest to the experiment. However, the slower behavior of the surge front in the experimental result could be due to a drag force between the fluid and the bottom which is neglected in the simulations when the free-sleep boundary condition is applied. As was expected, the results of the simulations using dynamic refinement are between the unrefined and the fully refined simulations for both the finer and the coarser discretizations. Despite that the results are very similar, it can be noticed that when using $(\epsilon, \alpha) = (0.55, 0.7)$ the results are closer to the results of the simulation with the fully refined domain than when using $(\epsilon, \alpha) = (0.5, 0.5)$.

4.1 Splash of a drop of water

For the simulations of the splash of a drop into rest water, the following situation is considered. A $10m$ high block of water is at rest in a $25m$ wide rectangular container, when a water drop with radius $2.0m$ with center at $(0m, 13.3m)$ and downward velocity of $2m/s$ starts moving until it splashes into the rest water. Like in the previous problem, no-slip boundary conditions are applied at the solid boundaries.

Four different simulations are performed using a strategy similar to that employed for the breaking dam problem to obtain different resolutions. For the simulation with the unrefined domain 360 particles discretize the rest water and 19 particles the water drop. In the fully refined domain the rest water is discretized by 1440 particles placed in a finer lattice and the drop is discretized by 76 particles. The simulations using dynamic

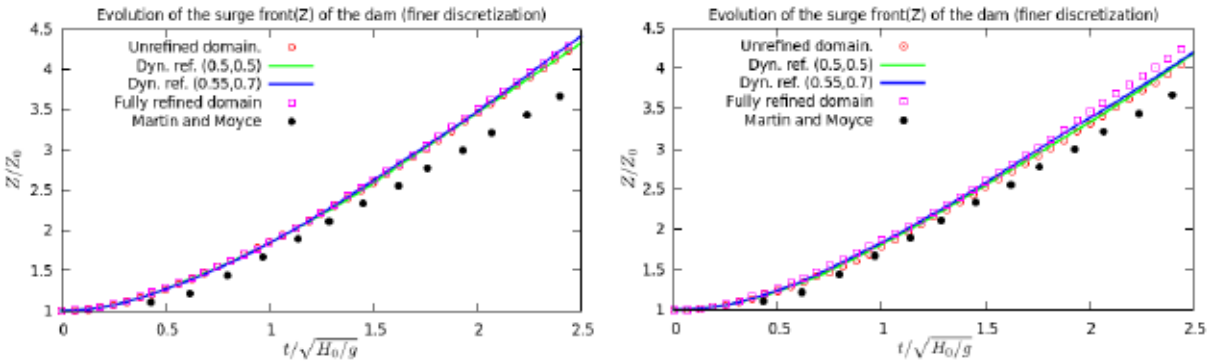

Figure 3: Evolution of the surge front of the braking dam simulation for the (a) finer discretization, and (b) coarser discretization.

refinement start from a discretization of the rest water as in the unrefined domain (360 particles), but using 76 particles to discretize the drop as in the fully refined domain. In the simulations using the dynamic refinement the velocity threshold of the refinement criterion was set to $1.5m/s$.

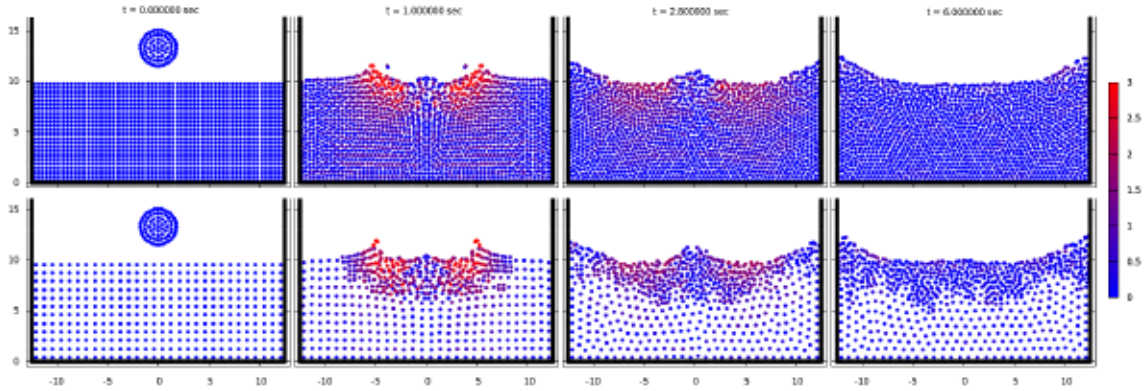

Figure 4: Simulation of the drop splashing into rest water.

Fig. 4 shows the evolution of the simulations with dynamic refinement using $(\epsilon, \alpha) = (0.55, 0.7)$ and with the fully refined domain. It can be seen that the movement of the surface as well as the distribution of the velocity are similar. Fig. 5 shows the evolution of the height of the water in each simulation. The simulation with dynamic refinement using $(\epsilon, \alpha) = (0.55, 0.7)$ reproduces very well the wavy behavior of the surface and gives a result that is very close to the result obtained with the simulation using the fully refined domain. Setting the dynamic refinement parameters (ϵ, α) equal to $(0.5, 0.5)$ leads to a substantial damping of the waves due to a damping of the particle velocities.

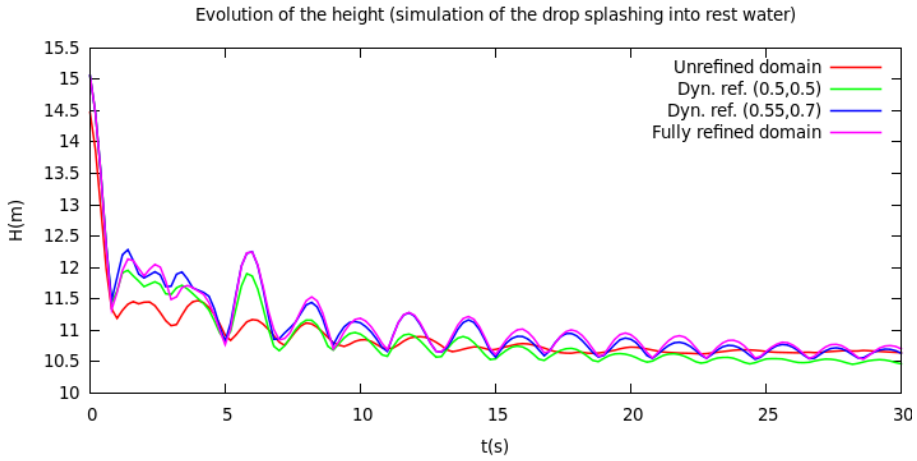

Figure 5: Evolution of the height (H) of the water in the simulation of the splash of the drop.

5 CONCLUSIONS

We presented a dynamic refinement algorithm for SPH. The refinement parameters, i.e. the separation parameter ϵ and the smoothing ration α are chosen such that the kernel gradient error is minimized.

The results for two model problems indicate that using the dynamic refinement procedure, instead of refining the whole domain, allows a substantial reduction in simulation time, while nearly the same accuracy is achieved.

Further issues to be studied are: a) dynamic coarsening, as the opposite of refinement; b) the implementation of the method in 3D; c) the application to other test cases.

ACKNOWLEDGMENT/DISCLAIMER: This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

REFERENCES

[1] Bate, M.R., Bonnell, I.A. and Price, N.M., Modelling accretion in protobinary systems, *Monthly Notices of the Royal Astronomical Society.* (1995) **277**:362–376.

[2] Chen, J. K. and Beraun, J. E., A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems, *Computer Methods in Applied Mechanics and Engineering.* (2000) **190**:225–239.

[3] Feldman, J., Dynamic refinement and boundary contact forces in Smoothed Particle Hydrodynamics with applications in fluid flow problems, *University of Wales Swansea, School of Engineering*, (2006).

[4] Feldman, J. and Bonet, J., Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems, *International Journal for Numerical Methods in Engineering.* (2007) **72**:295–324.

[5] Kitsionas, S. and Whitworth, A.P., Smoothed Particle Hydrodynamics with particle splitting, applied to self-gravitating collapse, *Monthly notices of the Royal Astronomical Society.* (2002) **330**:129–136.

[6] Kitsionas, S. and Whitworth, A.P., High-resolution simulations of clump-clump collisions using SPH with particle splitting, *Monthly notices of the Royal Astronomical Society.* (2007) **378**:507–524.

[7] Lastiwka, M., Quinlan, N. and Basa, M., Adaptive particle distribution for smoothed particle hydrodynamics, *International Journal for Numerical Methods in Fluids.* (2005) **47**:1403–1409.

[8] Liu, G. and Liu, M. *Smoothed particle hydrodynamics: a meshfree particle method*, World Scientific Publishing, (2003).

[9] Liu, M. and Liu, G., Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments. *Archives of Computational Methods in Engineering.* (2010) **17**:25–76.

[10] Martin, J.C. and Moyce, W.J., Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane, *Royal Society of London Philosophical Transactions Series A.* (1952) **244**:312–324.

[11] Meglicki,Z., Wickramasinghe, D. and Bicknell, G.V., 3D structure of Truncated accretion disks in close binaries, *Monthly Notices of the Royal Astronomical Society.* (1993) **264**:691–704.

[12] Monaghan, J.J., Smoothed particle hydrodynamics, *Reports on Progress in Physics.* (2005) **68**:1703–1759.

[13] Monaghan, J.J., Simulating free surface flows with SPH, *Journal of Computational Physics.* (1994) **110**:399–406.

[14] Monaghan, J.J. and Varnas, S., The dynamics of interstellar cloud complexes, *Monthly Notices of the Royal Astronomical Society.* (1988) **231**:515–534.

[15] Reyes López, Y. and Roose, D. Particle refinement for fluid flow simulations with SPH, *Computer Methods in Mechanics (CMM-2011),* Warsaw, Poland, 9-12 May (2011).